

Avances del horno SMD

Índice

1. 17-05-17	1
2. 18-05-17	2
3. 19-05-17	3
4. 21-05-17	5
5. 23-05-17	7
6. 24-05-17	7
7. 25-05-17	9
8. 26-05-17	9
9. 27-05-17	10
10.28-05-17	11
11.10-06-17	11
12.12-06-17	12
13.13-06-17	13
14.16-06-17	14
15.18-06-17	15
16.19-06-17	16
17.22-07-17	17

1. 17-05-17

Hoy estuve probando el control de tensión sobre la carga. El circuito que voy a utilizar es el de la figura 1.2. Luego de estar un rato teniendo problemas para que funcione, llegué a la conclusión de que el triac no es simétrico, y por ende el nodo donde se conecte el trigger es importante (ver figura 1.1).

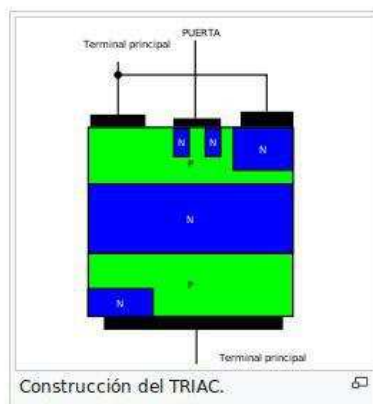


Figura 1.1: Construcción de un triac. Se puede ver que no es simétrico, y por ende importa el nodo donde se conecte el trigger.

En la figura 1.2 están anotados los pines del triac, ya que si se conecta de forma diferente no anda.

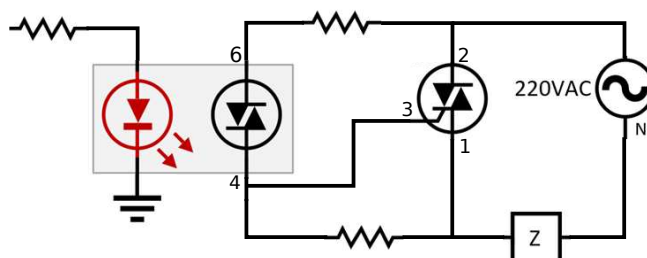


Figura 1.2: Circuito para controlar el triac.

Los valores de resistencias me parece que no importan demasiado, siempre y cuando no se supere las corrientes. Puse valores medio así nomás y siguió andando (en el que tengo armado ahora tengo una de 680Ω en la entrada del optotriac, dos de 100Ω en la salida del opto, y la carga es de aproximadamente 50Ω).

Estuve también probando un circuito detector de fase, ya que es necesario saber cuándo se cruza por cero para poder disparar el trigger correctamente. Senger me sugirió utilizar el circuito de la figura 1.3.

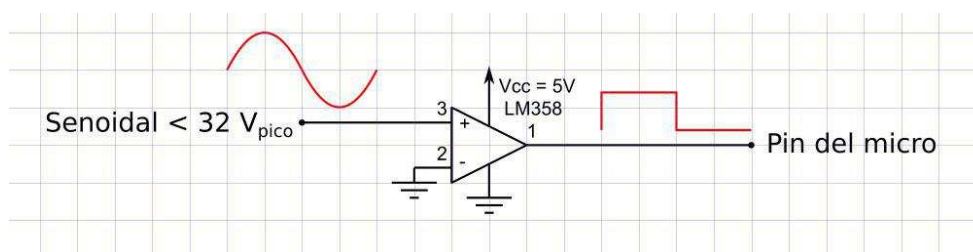


Figura 1.3: Circuito detector de cruce por cero propuesto por Senger.

El problema con el detector de cruce por cero de la figura 1.3 es que la senoidal de entrada no necesariamente está centrada en cero, ya que por ahora estoy probando con un trafo de 12v para no morir mientras pruebo, y la lógica la alimento desde mi nueva fuente de tensión. Por lo tanto, el único resultado que logré fue quemar un operacional cuando me equivoqué al conectar los cables. Mañana voy a probar otro circuito con un opto que debería funcionar.

2. 18-05-17

El circuito que voy a probar para detectar los cruces por cero es el de la figura 2.1. El opto que voy a usar es el FOD3180, ya que me sobraron un par del TP de circuitos 2 (siempre se quemaban de la nada, espero que no me pase lo mismo).

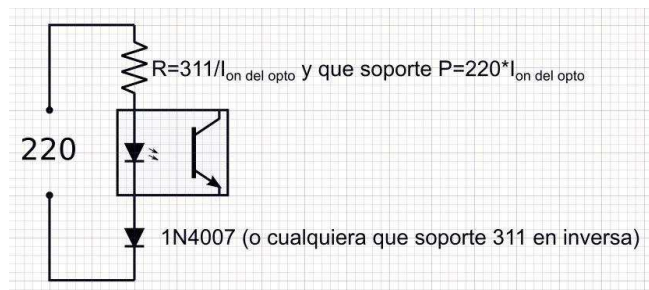


Figura 2.1: Circuito detector de cruce por cero.

Luego de quemar dos FODs (uno se quemó mágicamente, y el otro porque no se banco la tensión inversa), logré armar el circuito de la figura 2.2. Se puede ver que fue necesario agregar una resistencia de 100 kΩ en paralelo con la entrada del opto, ya que de forma contraria cae la mitad de la tensión en inversa en el semiciclo negativo (lo que quemó uno de los FODs). Con esta resistencia el diodo 1N4007 se banca toda la tensión en inversa.

A la salida del circuito de la figura 2.2 lo alimenté con 12V (con menos de eso no andaba), y conecté una carga de 2 kΩ a masa (la salida del FOD es CMOS, no es open collector). Al medir con el tester la salida, pude ver que la tensión era de 6V, lo cual en teoría me indica que está la mitad del tiempo encendido y la otra apagado. Esto no lo pude verificar, porque cuando quise medir frecuencia con el tester no lo pude hacer. Lo ideal sería usar un osciloscopio para ver que onda. Usé un Arduino para medir unos y ceros pero no tuve resultados muy informativos, y creo que es porque la salida no es tan digital como me gustaría (no lo probé demasiado porque al arduino no le gustaba demasiado estar conectado a la masa de la compu y la del circuito, y cada tanto se reseteaba).

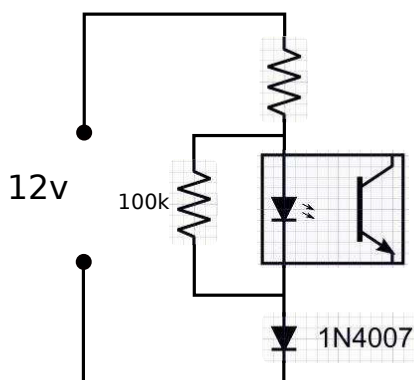


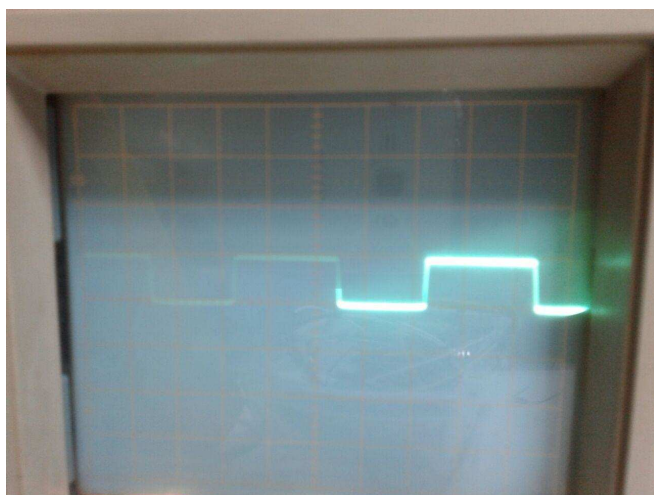
Figura 2.2: Circuito detector de cruce por cero que funcionó.

Hoy si no me da paja voy a ver de llevar las cosas y medir en la facu con el osciloscopio, y voy a pasar por microelectrónica a comprar optoacopladores que no sean tan maricones como los FODs.

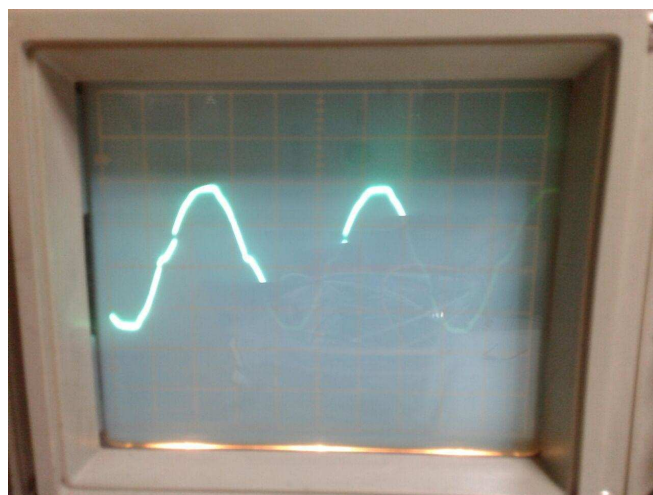
3. 19-05-17

Hoy me vine a lo de Senger a medir con el osciloscopio. Ayer compré los PC817, que son unos optos mas baratos y menos quemables que los FOD, pero tienen salida open collector, así que a la salida tuve que conectar una resistencia de pull-up.

En la figura 3.1 tengo los resultados de las mediciones con el osciloscopio.



(a) Salida del detector de cruce por cero. No es del todo simétrico porque estoy usando un trafo de 12v, pero asumo que cuando lo mande a 220v va a mejorar.



(b) Salida del triac, el cual se encuentra siempre disparado.

Figura 3.1: Mediciones con el osciloscopio de Senger. Las señales son muy lentas y por ende no pude sacar fotos muy copadas, pero alcanzan para afirmar que funciona.

En la figura 3.2 están los circuitos completos.

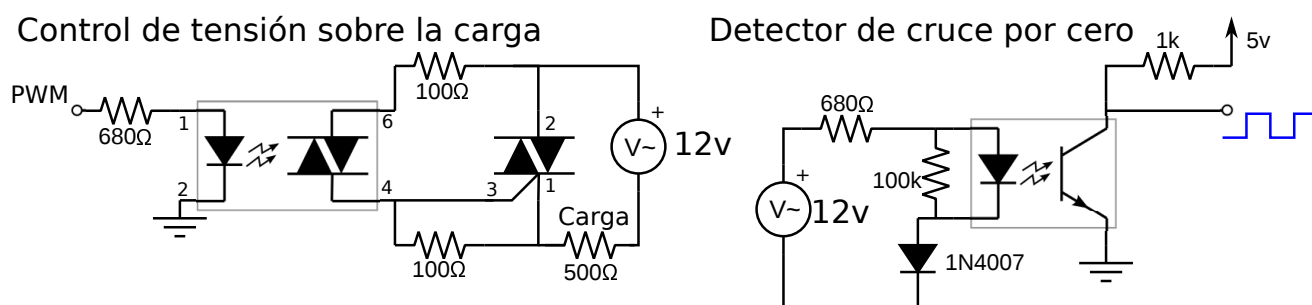


Figura 3.2: Circuito de control del triac y de detección de cruce por cero.

Ahora debería escribir un programita que me permita disparar el triac en distintos instantes para probar eso.

Escribí un programita re cabeza que salta con interrupciones de flancos ascendentes y descendentes en un pin, y lo que hace luego de eso es esperar un cierto tiempo y luego disparar el gate del triac. En la figura 3.3 tengo el resultado.

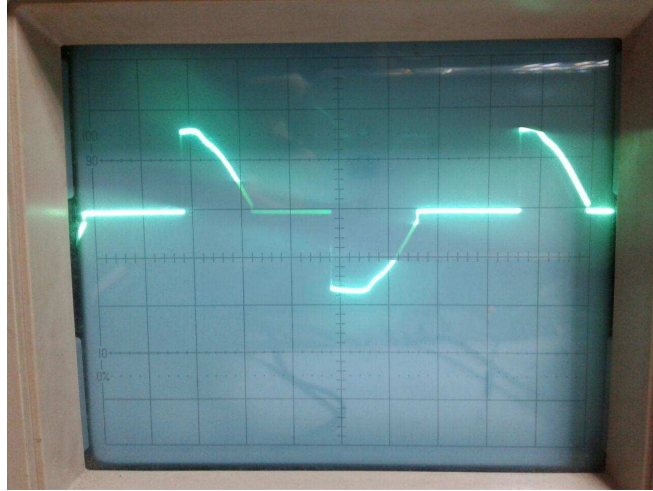


Figura 3.3: Salida del triac, el cual está siendo disparado por el micro, que a su vez utiliza la referencia de cruces por cero para sincronizarse con la senoidal.

El resultado obtenido era el esperado, incluso teniendo en cuenta que el delay para el disparo fue puesto a ojo. El siguiente paso es empezar a ver si se puede aplicar un PID que module la tensión de salida a partir de un pote por ejemplo. Además debería empezar a ver el tema de FreeRTOS, porque los delays debería ser lo más exactos posibles así evito pasarme de semiciclo y me quede el triac encendido.

Las cosas que debería encarar son:

- Ver el tema del firmware de la EDU-CIAA, que parece que es más copado.
- Analizar si conviene empezar a hacer un PCB, porque tengo todo en un proto re cabeza.
- Habría que hacer alguna prueba en 220V.
- Empezar a armar algo en FreeRTOS porque se nos viene la noche.
- Probar de nuevo la termocupla porque la probó Lio, y yo ni idea.
- Ya no se me ocurre mas nada, pero no tengo que colgar mas...

4. 21-05-17

Ahora que pude disparar el triac para recortar la senoidal, tengo que encontrar la relación entre el instante de disparo y la tensión eficaz de salida, ya que eso es lo que me va a permitir controlar la temperatura.

La señal que quiero analizar es:

$$v(t) = v_0 \sin(wt) \rightarrow \begin{cases} w = 2\pi 50\text{Hz} \\ v_0 = 311\text{v} \end{cases}$$

En la figura 4.1 esta la representación de lo que quiero hacer.

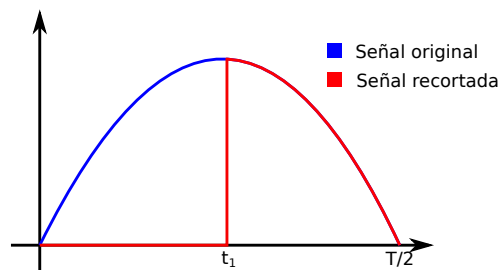


Figura 4.1: Representación de la señal recortada.

El valor eficaz se calcula como:

$$V_{ef} = \sqrt{\frac{1}{T} \int_{t_0}^{t_0+T} v(t)^2 dt}$$

Como yo voy a disparar el triac en cada semiciclo, en realidad tengo que integrar la mitad del período de la senoidal. Por lo tanto, integrando entre el instante en que disparo el triac y la mitad del período se tiene:

$$V_{ef} = \sqrt{\frac{1}{T} \int_{t_1}^{\frac{T}{2}} v(t)^2 dt}$$

Por lo tanto, puedo calcular

$$\begin{aligned} \int_{t_1}^{\frac{T}{2}} v(t)^2 dt &= v_0^2 \int_{t_1}^{\frac{T}{2}} \sin^2(wt) dt = v_0^2 \left(\frac{t}{2} - \frac{\sin(2tw)}{4w} \right) \Bigg|_{t_1}^{\frac{T}{2}} \\ &= v_0^2 \left(\frac{1}{2} \left(\frac{T}{2} - t_1 \right) - \frac{1}{4w} (\sin(Tw) - \sin(2t_1w)) \right) \\ &= v_0^2 \left(\frac{1}{2} \left(\frac{T}{2} - t_1 \right) + \frac{1}{4w} \sin(2t_1w) \right) \end{aligned}$$

El valor eficaz entonces es:

$$V_{ef} = v_0 \sqrt{\left(\frac{1}{4} - \frac{t_1}{2T} + \frac{\sin(2t_1w)}{8\pi} \right)}$$

Para $t_1 = 0$ el valor eficaz es:

$$V_{ef}(t_1 = 0) = v_0 \sqrt{\frac{1}{4}} = \frac{v_0}{2}$$

Por lo tanto, el porcentaje entre la señal recortada y la señal completa es:

$$\frac{V_{ef}(t_1)}{V_{ef}(t_1 = 0)} = \frac{v_0 \sqrt{\left(\frac{1}{4} - \frac{t_1}{2T} + \frac{\sin(2t_1w)}{8\pi} \right)}}{\frac{v_0}{2}} = \sqrt{1 - 2\frac{t_1}{T} + \frac{\sin(2t_1w)}{2\pi}}$$

Ahora tendría que despejar t_1 de esta expresión, ya que a mi lo que me interesa es saber en qué instante tengo que disparar el triac para tener un cierto porcentaje del valor eficaz máximo. Esto me parece que es imposible de hacer, así que voy a aproximar una relación lineal y chau. Una mejor forma sería armando una tabla de valores, pero eso lo voy a ver de hacer cuando ya tenga todo mas encaminado.

Usando una transformación lineal para sacar el valor del timer, y usando el ADC para medir un pote, logre modificar la tensión eficaz del triac moviendo el pote. La respuesta claramente no fue lineal, pero funciono bastante bien. Hay algunas fallas que no pude arreglar, pero creo que sin osciloscopio no voy a poder:

- El detector de fase se apaga antes de que la senoidal cruce por cero, ya que deja de polarizarse el opto. El problema con esto es que si el valor del timer es muy pequeño, puede que se dispare la señal de trigger antes de que se produzca el cruce por cero, y por ende en lugar de tener el 100 % del valor eficaz, se tiene casi un valor nulo (se prende con el timer y se apaga en el cruce por cero).
- El ancho del pulso lo meti a ojo, pero estaría bueno ver en el osciloscopio un valor más exacto.
- Si el valor del timer es muy grande, puede que el ancho del pulso haga que se solape con el siguiente cruce por cero. Ahora tengo puestos a ojo tanto un máximo como un mínimo del timer, pero tendría que medir para ver que sale.
- Estoy teniendo algunos líos con las interrupciones, porque si el timer lo configuro con un valor muy pequeño, creo que no me está dejando acceder al ADC, porque a veces queda como colgado y sin darle bola a las modificaciones en el ADC. Tendría que hacer un código mas presentable, o en lo posible empezar con FreeRTOS para gestionar bien las prioridades.

5. 23-05-17

Ayer estuve codeando todo en FreeRTOS, y recién hoy lo pude probar. Algo que me dí cuenta ayer, fue que la frecuencia del clock del micro estaba dividida por 4 (las librerías que había importado estaban así...), lo cual ahora se lo saqué para poder laburar sin estar justo de tiempos.

El código está disponible en mi repositorio de Github. No tengo un osciloscopio para ver lo que pasa, pero midiendo el valor eficaz de la salida me pareció que estaba andando todo bien: Usando un pote lograba regular la tensión eficaz de salida.

El problema que no pude solucionar todavía es el disparo del trigger: Para que el triac logre activarse, el trigger debe estar encendido un determinado tiempo. Lo que yo hice en un principio fue usar las funciones bloqueantes de FreeRTOS (`vTaskDelay` y `vTaskDelayUntil`) para hacer el delay, pero me parece que esto no funciona del todo bien porque explotaba (la salida quedaba en cero siempre). Reemplazando esto por un `while(i--)` bien cabeza para hacer el delay, funcionó todo perfectamente.

Otro problema que estuve viendo es que el micro se está colgando de la nada. Sin que se ejecute ninguna de las funciones de configuración, el micro salta a un default handler de la nada. Creo que tiene que ver con las interrupciones del micro, que tal vez cuando lo reseteo queda en un estado medio extraño, y el hecho de tener conectada la lógica para la detección del cruce por cero, hace que salte la interrupción externa, que como no se encuentra configurada hace cualquier cosa. Esto lo voy a revisar bien después, pero esto se podría solucionar simplemente haciendo que el detector de cruce por cero no funcione hasta que yo lo requiera, o simplemente desactivando las interrupciones en el primer paso del código (si es que el problema es ese...).

Otra cosa que hice hoy fue desarmar una fuente de 5v y 700 mA de un celular. Tenía todos los cables al aire, y por eso no la pude probar con los circuitos andando (no me daban las manos para sostener tantos cables). Mañana le voy a poner un cable mas copado, y así enchufarla a la pared. Igualmente medí que mis circuitos tienen un consumo de aproximadamente 250 mA, así que le puse una resistencia bien cabeza y medí que la fuente siguiera funcionando bien. Dado que la fuente pudo seguir regulando su tensión cuando le sacaba dicha corriente, asumo que la fuente está en buen estado y cuando la logre conectar bien, va a andar sin problema.

Por último, ayer empecé a hacer un esquemático para ver que onda el PCB. No estoy seguro si conviene hacer un poncho o una placa aparte, porque no me gusta la idea de tener la potencia cerca del micro.

Me voy a estudiar un poco de señales que la rindo la semana que viene, y ni empecé....

6. 24-05-17

Hoy decidí empezar a laburar un poco con el sensor de temperatura. Mañana capaz vaya a lo de Senger a laburar con el osciloscopio, así que voy a poder calibrar mejor lo del triac, y si termino lo del sensor de temperatura puedo empezar a ver de armar un PID.

El circuito que armó Lio, y que según él funcionó perfectamente, es el de la figura 6.1.

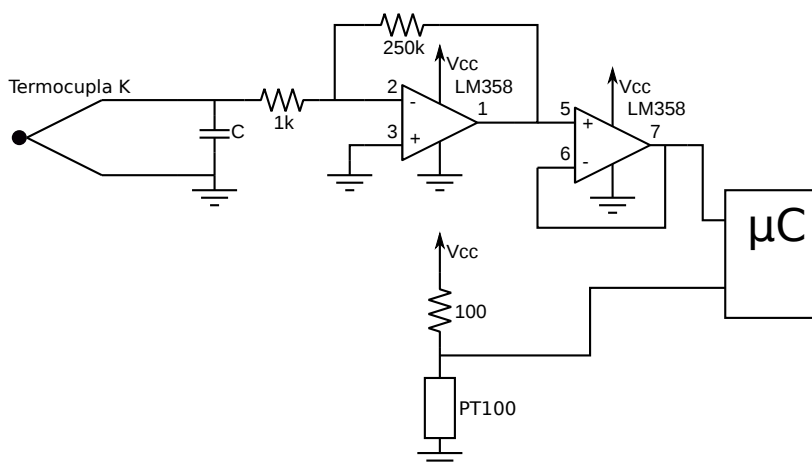


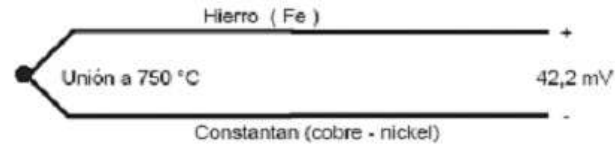
Figura 6.1: Circuito para medir la temperatura.

Acabo de probar el circuito de Lio con mi Arduino (Lio también había hecho un script para medir la temperatura con Arduino), y comparando los valores que me tira con los valores de la termocupla de mi tester, parece que todo funciona bien. Ahora voy a ver de pasar el código a la EDU-CIAA, pero seguro que eso funciona bien de una también (Grande Lio que lo dejó andando♡). El único problema que encontré con el circuito de Lio es que no tiene una temperatura de

referencia, y por lo tanto los valores que se miden se mueven bastante (el tester queda hiper clavado, porque seguro que tiene adentro algún sensorcito de referencia). Aprovechando que compramos un PT100, habría que ver cómo usarlo de temperatura de referencia para tener mediciones más precisas.

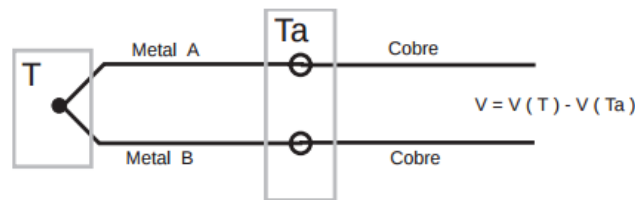
Termocuplas

Una termocupla se hace con dos alambres de distinto material unidos en un extremo, al aplicar temperatura en la unión de los metales se genera un voltaje muy pequeño, del orden de los milivolts el cual aumenta con la temperatura.



Midiendo la diferencia de potencial entre los bornes de la termocupla, se puede entonces conocer la temperatura.

El problema con este método, es que al conectar la termocupla a los terminales del circuito de medición, que son de cobre, se generan nuevas termocuplas.



La tensión que se mide a la salida de la termocupla entonces es:

$$v_{medida} = v_T - v_{Ta}$$

Donde v_T es la tensión de la termocupla, y v_{Ta} es la temperatura ambiente. Dado que lo que interesa es la tensión en la termocupla se tiene que despejar:

$$v_T = v_{medida} + v_{Ta}$$

Por lo tanto, midiendo la diferencia de potencial generada por la temperatura ambiente, es posible corregir la medición. Dado que la medición de la temperatura ambiente en este caso se va a hacer con un PT100, que no tiene relación con la termocupla, es conveniente aproximar todo de otra forma. En lugar de hacer la corrección con la diferencia de potencial, se puede hacer la corrección con las temperaturas:

$$T_{termocupla} = T_{medida} + T_{ambiente}$$

De esta forma, si se mide la temperatura ambiente con el PT100, y se le suma la temperatura medida con la termocupla, se va a poder obtener una medición de la temperatura bastante correcta.

Para conocer el mapeo de temperaturas de la termocupla se utilizó la tabla de <http://www.arian.cl/downloads/nt-002.pdf>. Usando la tabla de dicho documento y la función `polyfit` en Octave, se llegó a que:

$$T_{termocupla}(v) = 24599,766 \cdot v - 0,213$$

El PT100 es un sensor de temperatura que consiste en un alambre de platino que a $0^{\circ}C$ tiene 100Ω y que al aumentar la temperatura aumenta su resistencia eléctrica. El incremento de la resistencia no es lineal pero sí creciente y característico del platino de tal forma que mediante tablas es posible encontrar la temperatura exacta a la que corresponde (ver <http://www.arian.cl/downloads/nt-004.pdf>). Haciendo el mismo procedimiento que antes, es decir usando la tablita de valores para aproximar una recta, se obtuvo:

$$T_{PT100}(R) = 2,577 \cdot R - 257,774$$

La temperatura real entonces es:

$$T(v, R) = T_{termocupla}(v) + T_{PT100}(R)$$

Voy a escribir un pequeño código en el micro para probar esto.

Bueno, después de estar probando un par de cosas sin éxito, decidí ponerme a calibrar yo mismo los sensores. Los valores de la tabla esa que había encontrado no se ajustaban mucho con la realidad, y seguro que es porque incluyen los cambios por la temperatura ambiente, cosa que no puedo reproducir exactamente. Por lo tanto, empecé a tomar mediciones comparando con el termómetro de mi tester, y obtuve que:

$$\begin{cases} T_{termocupla}(v) = 23376,525 \cdot v - 1,574 \\ T_{PT100}(R) = 2,857 \cdot R - 239,178 \end{cases}$$

El tema con esta calibración, es que implícitamente estoy calibrando la termocupla para la temperatura ambiente actual, y por ende cuando mueva el sistema a otro lugar, la calibración va a dejar de servir. La solución sería usar el PT100 como referencia, pero dado que ambos sensores me miden siempre lo mismo, si sumo las mediciones me pasa a dar cualquier cosa. Por esta razón creo que voy a usar simplemente la termocupla, y jugarmela a tener un poco de error en la medición. Si después veo que se hace necesaria una medición mejor, veré de incluirlo (no es mucho laburo tampoco).

Mañana me voy a lo de Senger a medir el tema del triac. Tal vez aproveche y lo mande a 220V para ver que onda. Ahora me voy a tomar un cafecito, descansar un ratito, y ponerme con señales que ayer pude aprender unas cuantas cosas (o tal vez me juege un Age...).

7. 25-05-17

Hoy vine a lo de Senger a laburar con el osciloscopio, pero lamentablemente no pude avanzar nada, ya que me convencieron para jugar una partida de Age de como tres horas...

Estuve viendo de generar el ancho del pulso de trigger mediante una interrupción del timer, para sacar el `while()` cabeza que tenía. Esto no lo pude lograr por los siguientes motivos:

- En la tarea que dispara el trigger, para evitar que me saquen la CPU durante el disparo (y así evitar que quede encendido demasiado tiempo), tuve que entrar en modo crítico, es decir que el scheduler queda desactivado. El problema es que esto también me deshabilita las interrupciones para el TIMER, y por lo tanto no puedo controlar el delay de esta forma.
- Si yo decido sacar la parte del modo crítico, me sacan la CPU en la mitad del proceso, y hace que no funcione del todo bien. Esto tal vez lo pueda mejorar agrandando el time slice, pero eso va a provocar que haya mas demoras en otras tareas (actualmente esta el time slice en 1 ms supuestamente).

Dados estos inconvenientes, creo que voy a dejar el `while()` como retardo para el trigger y voy a seguir avanzando con las demás cosas, total no es algo demasiado esencial.

Algo que no llegué a probar, fue de conectar a 200v para ver como funciona todo (tendría que cambiar todas las resistencias de mis circuitos), y además creo que eso va a mejorar el funcionamiento del detector de cruce por cero, ya que van a ser mas despreciables los delays de encendido y apagado del detector.

El próximo paso creo que va a ser conectar todo a 220v y ver cómo funciona, y después conectarle una resistencia para empezar a calentar cosas.

8. 26-05-17

Hoy intenté de nuevo hacer el delay que mantiene encendido el trigger con un timer, y esta vez sí funcionó. Lo hice bien de a poco, probando cada modificación, y lo dejé funcionando perfectamente. Para que funcione tuve que eliminar la parte que entra en modo crítico, porque sino las interrupciones no se atienden hasta que se sale de la tarea. No sé qué es lo que estaba haciendo mal ayer, pero lo importante es que anda.

También modifiqué la frecuencia de muestreo del ADC. Como las variaciones de temperatura son lentas, puse una variable que simplemente cuenta los cruces por cero, y cuando se llega a un valor máximo se actualiza el valor del ADC. El valor máximo lo voy a tener que definir cuando empiece a sensar temperatura, así que por ahora lo dejé en un valor cualquiera.

También estuve modificando algunas cosas del esquemático del PCB. Agregué una bornera para el PT100, por si decido usarlo, cambié todas las resistencias y capacitores a SMD, y definí algunos valores de componentes. Nada importante.

Ahora voy a mirar si la resistencia que tengo para calentar está quemada, y ver si cuando mando todo a 220v sigue andando bien.

Primera conexión a 220

Para poder hacer esto, tuve que cambiar los valores de resistencias para evitar que se me quemen los optos. Sin embargo, tenía mal rotuladas unas resistencias de 47Ω como de $47\text{ k}\Omega$, y por ende voló el opto del detector de fase. Una vez corregido esto volví a conectar a todo, pero lamentablemente me dí cuenta tarde que no había cambiado las resistencias que funcionaban como carga. El resultado: Un fogonazo como jamás pensé que podían tirar las resistencias de carbón.

Finalmente, luego de tener un par de caídos en el camino, pude conectar a 220v una carga de $100\text{ k}\Omega$. Moviendo el potenciómetro pude regular bastante la tensión sobre la carga, pero tuve problemas con los extremos del pote, ya que tenía hardcoded los límites del timer para 12v, pero seguro que ajustándolos para los que yo quiero hacer ya va a andar todo bien.

El próximo paso va a ser conectar todo a la resistencia calentadora y empezar a medir su cambio de temperatura. Tal vez me convenga ir haciendo el PCB para evitar estar tan expuesto a los 220v, pero lo voy a ver en otro momento. Ahora me tengo que preparar para dar una presentación de intro a proyectos, la mejor materia del planeta.

9. 27-05-17

Hoy empecé a calentar cosas de una vez por todas. Quise calibrar un poco mejor el tema del timer, porque lo tenía calibrado para los 220v, pero no tuve buenos resultados. Creo que esto va a ser mejor hacerlo con el osciloscopio.

Hoy vino Male ♡ y me estuvo dando una mano. Lo que hicimos fue meter el calentador de agua en un frasco, y medir la respuesta al escalón. No fue una medición muy precisa, pero creo que sirve para estimar los parámetros del PID. Después de estimarlos voy a ver si puedo cerrar el lazo.

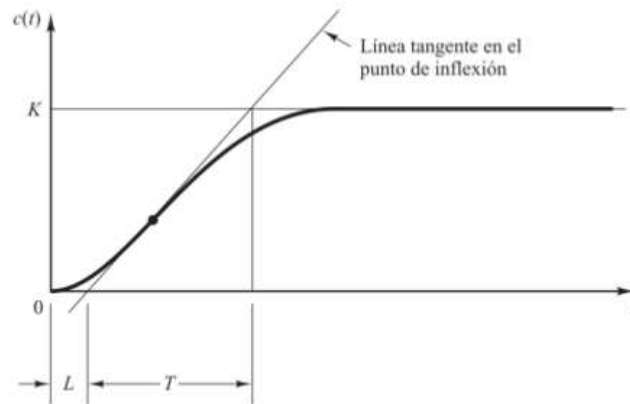


Figura 9.1: Curva para estimar los parámetros del PID mediante Ziegler-Nichols.

Los parámetros estimados de la resistencia a lazo abierto son:

$$\begin{cases} K = 72^\circ\text{C} \\ L = 1\text{s} \\ T = 250\text{s} \end{cases}$$

Con estos parámetros, las constantes del PID son:

$$\begin{cases} k_p = 1,2 \cdot \frac{T}{L} = 300 \\ T_i = 2L = 2 \\ T_d = \frac{L}{2} = \frac{1}{2} \end{cases} \rightarrow C(s) = k_p \left(1 + \frac{1}{T_i s} + T_d s \right) = 300 + \frac{150}{s} + 150s$$

Por lo tanto, calculando la diferencia entre la salida real y la deseada, se debe hacer:

$$y_{ref} - y(t) = e(t) \rightarrow u(t) = 200e(t) + 150 \int e(t) dt + 150\dot{e}(t)$$

Aplique todas estas cosas al código de prueba de FreeRTOS, donde agregué una nueva tarea que aplica el PID cada vez que se le entrega un semáforo (el ADC se lo otorga), y creo que tuve buenos resultados. Le estuve harcodeando una temperatura de referencia, y cuando se llegaba a una temperatura similar el circuito se apagaba (lo medi con mi tester, pero no siempre miden lo mismo las termocuplas).

El problema que estoy teniendo es el siguiente: Con la termocupla mido temperatura, y la comparo con la temperatura de referencia. Sin embargo, una vez que aplico el PID, tengo que mapear esa salida con el delay del timer que dispara el trigger. Lo que hice básicamente fue mapear de forma medio lineal entre los valores extremos que puede tomar el delay, pero no se si es la mejor forma porque no alcance la temperatura de referencia de forma exacta. En estos días voy a ver de solucionarlo, pero ahora estoy muy cansado. Lo bueno es que ya estoy calentando agua, y es probable que ajustando un poco el controlador ya tenga la parte mas importante del tp andando.

10. 28-05-17

Hoy seguí viendo el tema del controlador. Descubrí que por la termocupla me está entrando ruido de baja frecuencia proveniente de los 220v. Senger me recomendó que pusiera un pasabajos en el circuito que acondiciona la señal de la termocupla, así que puse un par de capacitores en la primer etapa del circuito de la figura 6.1, en paralelo con la resistencia de 250 k Ω . Esto hizo que anduviera mejor, pero sigue habiendo muchas oscilaciones en la medición de la termocupla. Supongo que por culpa de estas oscilaciones el PID no termina de funcionar correctamente, pero todavía no pude comprobar nada de forma concluyente. Mañana voy a seguir viendo que onda, pero me parece que me conviene hablar con algún profesor para ver si lo estoy encarando bien.

11. 10-06-17

Todos estos días estuve laburando medio por arriba porque tenía que rendir señales. La otra vez había hecho el PCB, pero las pistas me quedaron muy finitas, y por eso andaba medio rancio. Así que tuve que volver a hacer el PCB, y ahora está andando bien. En el repositorio de Github está todo.

Lo que tengo que hacer ahora, es muestrear la curva de temperatura, y así ir siguiéndola.

En la figura 11.1 tengo el resultado obtenido. Lo que hice fue escalar el perfil para poder seguirlo con la resistencia que tengo, y actualice la temperatura cada 30 segundos. En el gráfico se muestran mediciones de la temperatura cada 3 segundos. Se puede ver que los resultados son parecidos a los deseados, con la diferencia de que no es posible reducir la temperatura ya que es imposible enfriar el agua. Se supone que en el horno se va a poder hacer sin problemas. Falta calibrar la termocupla, porque al parecer cuando la atornillo a la placa, se forma una termocupla diferente a la que tenía antes, entonces la calibración de antes ya no sirve.

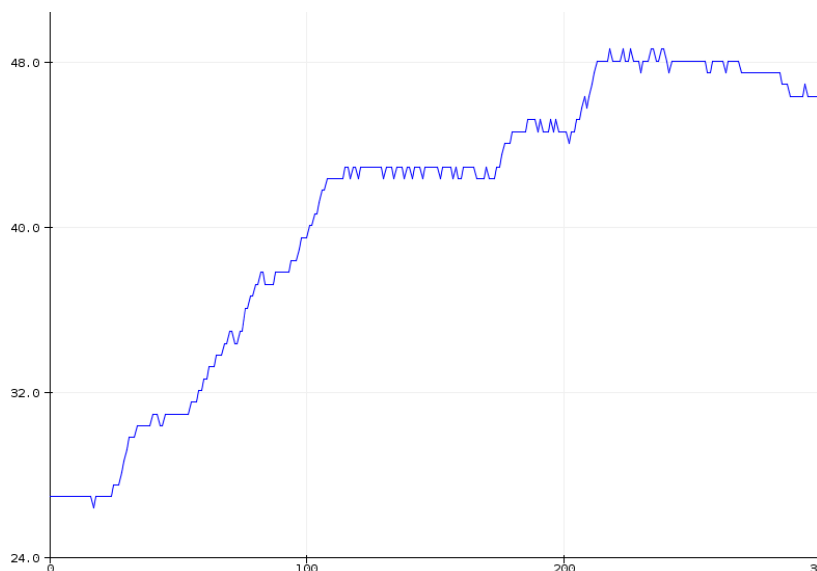


Figura 11.1: Perfil de temperatura obtenido.

12. 12-06-17

Ayer desarmé el horno y encontré que de las 4 resistencias que tiene sólo estaban quemadas dos. El problema es que están conectadas de a dos en serie y después en paralelo, entonces se habían quemado justo dos de distintas ramas y había dejado de andar. Simplemente puse en serie las dos resistencias que andaban y saqué afuera del horno los cables de alterna, y dos cables para la resistencia.

En la figura 12.1 están los resultados obtenidos con el horno.

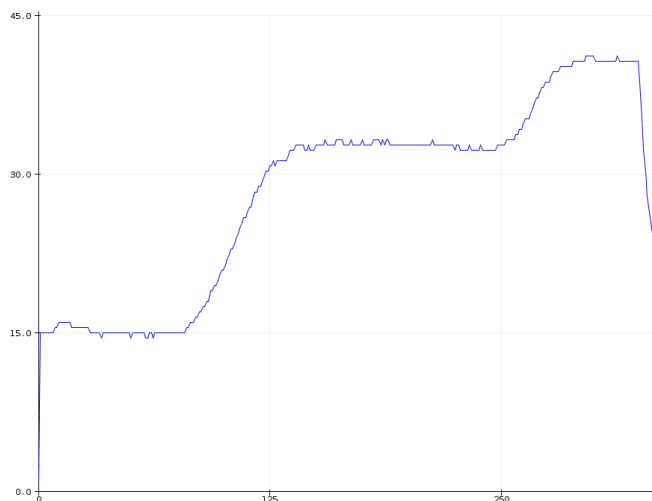


Figura 12.1: Resultados obtenidos con el horno. La caída abrupta en la temperatura al final de la curva se debe a que abrí la puerta. Dado que no puse el ventilador, no es posible bajar la temperatura rápidamente.

El problema con el resultado obtenido es que la inercia térmica del horno es gigante. Por lo tanto cuando apago la potencia la temperatura sigue subiendo, de forma tal que se superan los valores de referencia que le siguen, y por lo tanto la potencia permanece apagada mucho tiempo, provocando que no se pueda seguir de forma precisa la temperatura deseada.

Tendría que ver de armar de nuevo el código con el PID, y ver si con eso mejora, pero va a ser complicado meter todo eso dentro del código de ahora.

Seguí probando algunas cosas, y obtuve los resultados de la figura 12.2. Lo que hice fue aumentar la temperatura, bajar el tiempo de actualización entre muestras, y acerqué la termocupla a las resistencias para reducir la deriva térmica. Si bien mejoró un poco el tema de la inercia térmica, es mucho mas apreciable el ripple de la respuesta. Tengo que ver como sigo, pero me parece que el PID que había diseñado antes tampoco me va a servir, porque el sistema respondía tan lento que el error acumulado hacía que se disparara el trigger, por lo que terminaba siendo casi igual a un controlador on-off. Voy a ver de meter el ventilador dentro del horno, y así ver si se homogeniza la temperatura, y se puede reducir un poco la inercia.

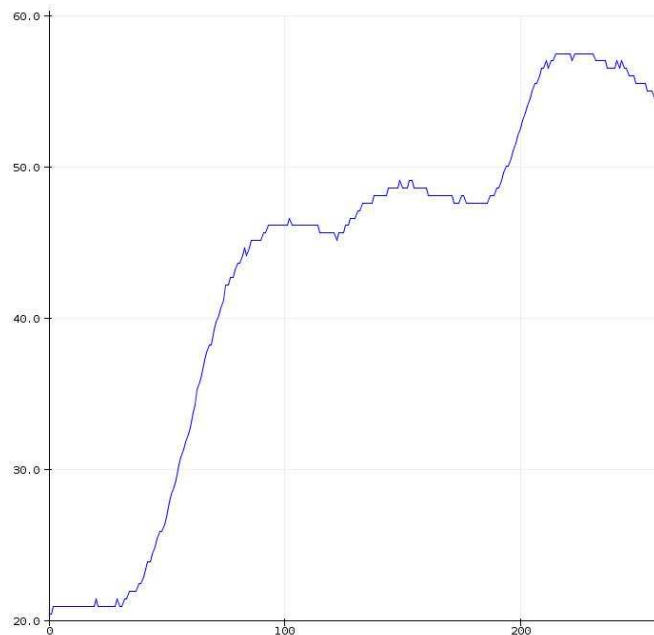


Figura 12.2: Resultados obtenidos con el horno.

En la figura 12.3 están unas fotos de la placa que armé.

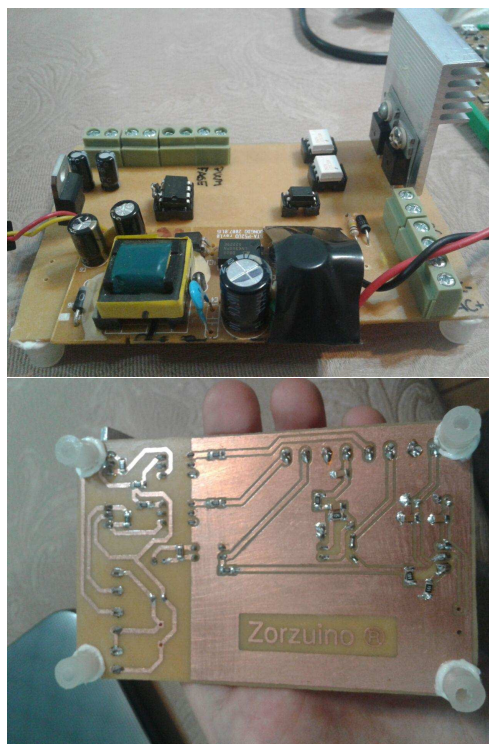


Figura 12.3: Fotos de la placa armada. La fuente de 220 – 5V de celular de la placa no está actualmente conectada.

13. 13-06-17

Hoy cambié el código para usar un controlador proporcional que controla la cantidad de semiciclos enteros que permanece encendida la potencia. Definí una macro que dice la cantidad máxima de ciclos a contar (en este caso puse 10), y en función del valor que tira el controlador se enciende la potencia una determinada cantidad de ciclos, que se obtiene

como porcentaje del máximo. Ésto permitió tener una respuesta mucho más suave y precisa. En la figura 13.1 están los resultados.

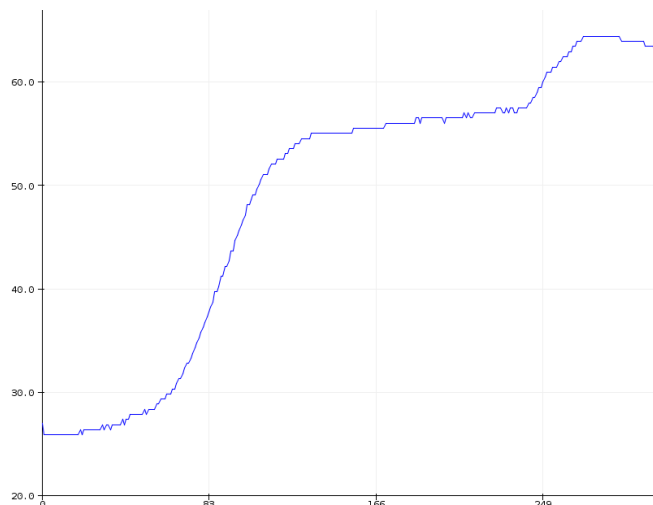


Figura 13.1: Control proporcional por ciclos enteros aplicado en el horno.

Faltaría el tema de bajar la temperatura, para lo cual se hace necesaria la utilización del ventilador. El problema es que tengo dudas de que el ventilador esté aislado, entonces si lo meto en el horno puede ser peligroso.

Después de muchos días de no saber cuál era el problema, logré solucionar el tema con los botones de encendido y apagado (por alguna razón sólo podía prender y apagar un par de veces, y después se colgaba). El problema era que la tarea que actualizaba la referencia de temperatura usaba la función `vTaskDelayUntil`, la cual recibe como argumento el último tiempo de encendido, se volvía loca cuando apagaba la placa. La cambié por la función `vTaskDelay` que no requiere dicho argumento, y no tuve mas el problema. La desventaja es que las actualizaciones no son tan precisas, pero no vi que se afectaran los resultados.

Recién soldé la fuentecita de celular a la placa, cosa de alimentar todo directamente de la alterna. El problema que tuve es que la temperatura la estoy midiendo con un Arduino, y entonces se rompe todo cuando uno la masa de la compu con la masa del circuito. Ahora volví a desoldar la fuente, y lo voy a dejar así estos días, total no cambia demasiado. Además se supone que Lío va a medir la curva y mandarla por Bluetooth, así que si lo termina no va a hacer falta el Arduino.

14. 16-06-17

Vinieron Lio y Senger y estuvimos avanzando. Lio arrancó con su parte, y yo estuve instalando el ventilador en el horno. El ventilador simplemente funciona a una velocidad baja durante la etapa de calentamiento, y se enciende a máxima velocidad cuando se terminó con el ciclo de temperatura. En la figura 14.1 está el resultado obtenido. El resultado es un poco más ruidoso (el Arduino muestrea más rápido y sin promediar que la EDU-CIAA, por lo que es más ruidoso que lo que ve el controlador) pero se sigue bien la curva. El sensor lo puse bastante cerca de las resistencias, pero tal vez midiendo más en el centro del horno se ve algo más suave.

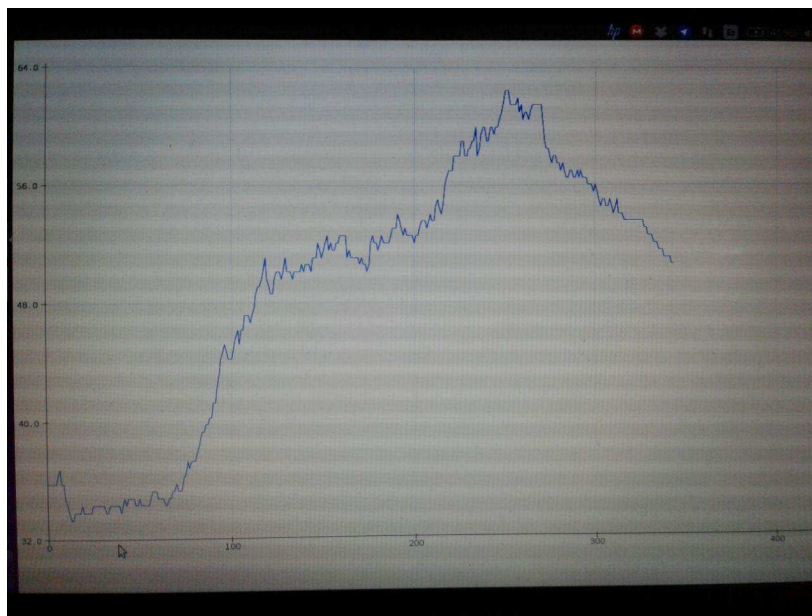


Figura 14.1: Resultados obtenidos con el ventilador.

Compramos 4 resistencias para el horno, así que mañana las voy a colocar y ver si logro llegar a la temperatura necesaria para hacer soldaduras.

15. 18-06-17

Hay buenas y malas noticias. Ayer puse las 4 resistencias nuevas, poniendo 2 en serie en paralelo con otras dos en serie. El problema con estas resistencias es que son de una potencia mucho mayor que las viejas, y entonces el triac no es capaz de manejar tanta corriente. El primer parche fue poner otro triac en paralelo y ver si mejoraba, pero al aumentar más la temperatura seguía explotando todo (cuando aumenta la temperatura, el valor de las resistencias se reduce y aumenta más la corriente. Si la corriente es superior a la que el triac puede manejar, parece que queda siempre en conducción, lo que provoca mayor corriente y mayor temperatura). Intenté poniendo un tercer triac en paralelo pero la corriente de trigger es muy pequeña para encender los tres, así que le dejé únicamente dos en paralelo.

También probé distintas combinaciones con las resistencias: las 4 en serie, 3 en serie, dos en serie, 2 en paralelo en serie con otra, etc. Si bien estas combinaciones funcionaban, no permitían subir la temperatura hasta los 250°C.

Actualmente estoy usando el BT137, que tiene una corriente de operación de 6A, lo cual está medio al límite. Mañana (Lunes) voy a ir a comprar el BT139 o alguno similar, que se banca como 10A o 12A.

Algo que acabo de probar, fue de poner dos resistencias nuevas en serie con las dos viejas que andaban, y obtuve los resultados de la figura 15.1. Se puede ver que pude llegar hasta los 200°C (en realidad hasta los 216°C, ya que el Arduino tiene una diferencia con la medida real). Sin embargo para poder hacer esto tuve que duplicar el tiempo de establecimiento entre muestra y muestra, es decir que toda esta curva duró el doble que la curva deseada (en lugar de durar dos minutos y medio, duro cinco). El otro inconveniente fue que cuando el micro cortó la potencia al finalizar el recorrido de toda la curva, sobre la resistencia seguía habiendo tensión, lo cual indica que el triac quedó clavado en conducción.

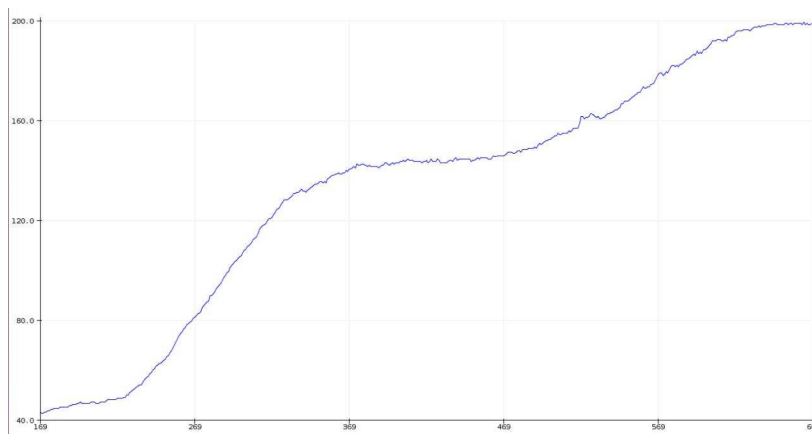


Figura 15.1: Resultados obtenidos combinando las resistencias viejas con las nuevas.

Creo que no queda otra que esperar hasta mañana y comprar triacs mas pulentas, porque con los que tengo dudo que pueda lograr resultados mejores. Lo bueno de todo esto es que el control de temperatura funciona muy bien, y que solucionando el tema del control de potencia, ya estaría andando todo bien.

En la figura 15.2 están las fotos de las villadas que tuve que hacer por no tener triacs mas grosos.

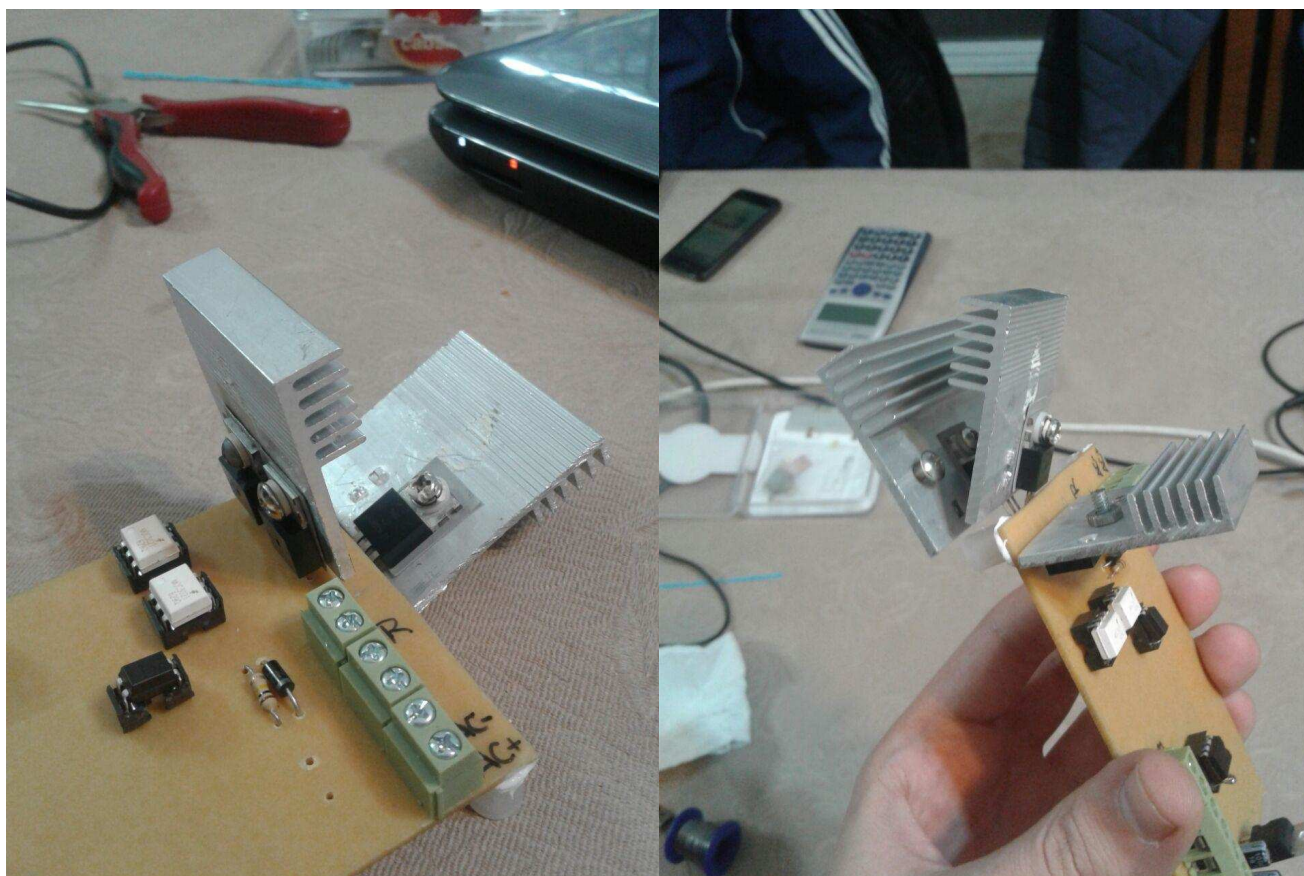


Figura 15.2: Fotos de los triacs en paralelo. A la izquierda se tienen 2 triacs en paralelo, y a la derecha se tienen 3. El caso con 3 triacs en paralelo no funcionó ya que la corriente de trigger era pequeña.

16. 19-06-17

Hoy compré el BTA24, que es un triac de 25A, para reemplazar el BT137. Puse las 4 resistencias nuevas, y salió andando todo como piña. Una vez armado todo, se procedió a soldar. Pusimos el estaño en pasta que me habían regalado en la CNEA, y lo mandamos al horno. En la figura 16.1 están los resultados del día de hoy.



(a) Primer soldadura realizada.



(b) Segunda soldadura realizada. Se puso estaño en pasta sobre el nombre en cobre para rellenarlo.

Figura 16.1: Resultados logrados con la primer soldadura.

Al parecer el horno ya está funcionando bien como para un primer prototipo. El único problema que tuvimos fue que las placas se tostaron un poco, pero el estaño se derritió bien.

Hoy también probamos al aplicación de celular que hizo el amigo de Lío, y mañana va a volver para probar todo junto. Parece que todo viene encaminado, pero se entrega pasado mañana, así que mejor meterle pata.

17. 22-07-17

Ayer me levanté temprano y arreglé todos los problemas que teníamos con la interconexión entre la placa de Lio y la mía, y quedó todo andando. Presentamos el TP y pudimos aprobarlo. Algo que había faltado era una foto de la aplicación de celular que hizo el amigo de Lio, lo cual está en la figura 17.1.

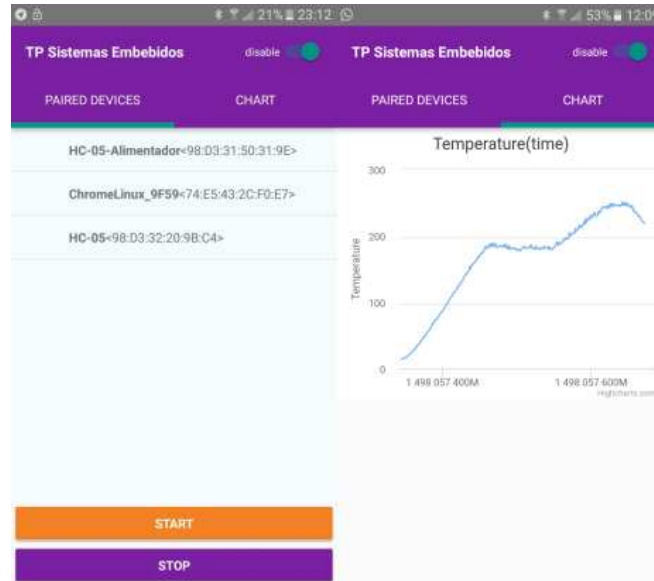
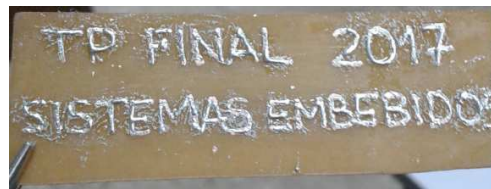


Figura 17.1: Foto de la aplicación de celular.

Por lo tanto, ya puedo despedirme de este trabajo, donde pude aprender muchas cosas y me entretuve un buen rato.
Saludos cordiales,
Nicolás Zorzano.

FIN DE TRANSMISIÓN.



Referencias

- [1] Link de Github al proyecto: https://github.com/nicozorza/TP_Final_Embebidos
- [2] <http://www.e-ika.com/control-de-potencia-con-arduino>
- [3] <http://voltiosybytes.blogspot.com.ar/2014/02/control-de-potencia-en-ac-con-triac-y.html>